UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/731,617	12/07/2000	James Allan Kahle	AT9-99-445	1084

7590 11/05/2003

Joseph P. Lally
DEWAN & LALLY, L.L.P.
P.O. Box 684749
Austin, TX 78768-4749

EXAMINER	
GOLE, AMOL V	
ART UNIT	PAPER NUMBER
2183	4

DATE MAILED: 11/05/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	**Application No.** 09/731,617	**Applicant(s)** KAHLE ET AL.	
	Examiner Amol V. Gole	**Art Unit** 2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE _3_ MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on _12/07/00;04/16/01_ .

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) _1-19_ is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) _1-19_ is/are rejected.

7)☒ Claim(s) _8,9,15 and 16_ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on _16 April 2001_ is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

11)☐ The proposed drawing correction filed on _____ is: a)☐ approved b)☐ disapproved by the Examiner.

If approved, corrected drawings are required in reply to this Office action.

12)☐ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. §§ 119 and 120

13)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____ .

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

14)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).

a) ☐ The translation of the foreign language provisional application has been received.

15)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s) _____ .
4) ☐ Interview Summary (PTO-413) Paper No(s). _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: .

DETAILED ACTION

1. Receipt is acknowledged of the following papers:

1) 4 sheets of substitute drawings as requested

These papers have been placed of record in the file.

2. Claims 1-19 have been examined.

Drawings

3. Figures 1-3 should be designated by a legend such as --Prior Art-- because only that which is old is illustrated. See MPEP § 608.02(g). A proposed drawing correction or corrected drawings are required in reply to the Office action to avoid abandonment of the application. The objection to the drawings will not be held in abeyance.

4. The drawings are objected to as failing to comply with 37 CFR 1.84(p)(5) because they do not include the following reference sign(s) mentioned in the description: 400, 402, 404. A proposed drawing correction or corrected drawings are required in reply to the Office action to avoid abandonment of the application. The objection to the drawings will not be held in abeyance.

Specification

5. The abstract of the disclosure is objected to because the title disclosed in the abstract ("Non-stall Pipeline Microprocessor") is different from the title of the application. Correction is required.

6. The abstract of the disclosure is objected to because the abstract may not exceed 150 words in length. Correction is required. See MPEP § 608.01(b).

7. The title of the invention is not descriptive. A new title is required that is clearly indicative of the invention to which the claims are directed.

The following title is suggested: HARDWARE SUPPORT FOR DUAL PATH BRANCH EXECUTION IN RESPONSE TO COMPILER GENERATED BRANCH PREDICTION INFORMATION IN A CONDITIONAL BRANCH INSTRUCTION.

8. The disclosure is objected to because of the following informalities:

1) In the "Brief Description of Drawings" section, mention that figures 1-3 are prior art.

2) On pg. 8, line15, the word "results" introduces grammatical errors.

3) On pg. 10, line 16, place a '.' to mark the end of the sentence after the words " 5-bit secondary opcode field 404".

Appropriate correction is required.

9. The specification is objected to as failing to provide proper antecedent basis for the claimed subject matter. See 37 CFR 1.75(d)(1) and MPEP § 608.01(o). Correction of the following is required: Claims 12 and 13 recite the limitation "depending upon the state of the branch instruction information". There is insufficient antecedent basis for this limitation in the claims.

Claim Objections

10. Claim 8 is objected to because it recites the limitation "the branch prediction unit" in line 1. There is insufficient antecedent basis for this limitation in the claim.

11. Claim 9 is objected to because it recites the limitation "the branch prediction information" in line 2. There is insufficient antecedent basis for this limitation in the claim.

12. For the purposes of the following art rejection, "the branch prediction information" is being interpreted as the branch prediction information from the branch prediction unit.

13. Claims 9 and 16 objected to because of the following informalities: On line 1 of both the claims the word "prediction" is spelled incorrectly as "predicution". Appropriate correction is required.

14. Claims 15-19 are objected to because they refer to "the microprocessor " of claim 14 while claim 14 discloses a "processor".

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

15. Claims 1-19 are rejected under 35 U.S.C. 103(a) as being unpatentable over Sharangpani et al. (US005860017A) in view of Maki (US005729707A).

16. In regard to Claim 1:

17. Sharangpani et al. teach a method of executing instructions in a microprocessor (col. 16, line 47) comprising:

1) fetching a conditional branch instruction (col. 16, line 53) from an instruction cache (col. 5, lines 35-37);

2) predicting the branch outcome and determining whether the prediction will be accurate or not (col. 16, lines 54-57); and

3) responsive to the branch prediction information (col. 16, lines 60-62), fetching instructions from both a branch-taken path and a branch-not-taken path of the branch instruction (col. 16, lines 59-60).

18. Sharangpani et al. do not disclose the detection of branch prediction information in the branch instruction as claimed in the instant application but instead generates the branch prediction information by performing branch prediction in hardware on fetching the conditional branch instruction.

19. Maki teaches a prediction means (col. 12, line 65), which detects (extracts, col. 12, line 65) branch prediction information (branch prediction bit, col. 12, lines 65-66) in the branch instruction.

20. One of ordinary skill in the art at the time of the invention would have recognized that the dynamic branch prediction (Fig. 3, element 316) method disclosed by Sharangpani et al. could be replaced by the static-prediction means taught by Maki and hence reduce hardware costs.

21. Thus it would have been obvious to one of ordinary skill in the art at the time of the applicant's invention to modify the method disclosed by Sharangpani et al. by implementing the prediction means taught by Maki to reduce hardware costs.

22. In regard to Claim 2:

23. Sharangpani et al. further teach of:

1) speculatively executing the instructions from the branch-taken path and the branch-not-taken path of the branch instruction (Fig. 6B, element 627, col. 14, lines 39-43);

2) executing (resolving, col. 17, line 37) the conditional branch instruction;

3) based upon the outcome of the conditional branch instruction, discarding the results from the speculatively executed instructions from the branch-taken path if the branch is not taken and discarding results from the branch-not-taken path if the branch is taken (col. 17, lines 38-41, 46-47).

24. In regard to Claim 3:

25. Sharangpani et al. do not disclose that the branch prediction information comprises of compiler generated information indicative of the context in which the conditional branch instruction is used.

26. However Maki teaches that the branch prediction information (branch prediction bit, col. 9, line 24) is prepared according to the known prediction algorithm when a compiler makes a program (col. 9, lines 25-28).

27. One of ordinary skill in the art at the time of the invention would have recognized that by using the branch prediction information comprising of compiler generated information indicative of the context in which the conditional branch instruction as taught by Maki in place of the dynamic branch prediction (Fig. 3, element 316) method taught by Sharangpani et al. would have reduced costs by reducing prediction logic in hardware.

28. It would have been obvious to one of ordinary skill in the art at the time of the applicant's invention to have used branch prediction information comprising compiler generated information as taught by Maki for the branch prediction disclosed in Sharangpani et al. to reduce hardware costs.

29. In regard to Claim 4:

30. Sharangpani et al. disclose that the branch prediction information causes instruction fetching from both the taken and not taken branches (col. 16, lines 59-60), if the branch instruction is determined (resolution of the condition, col. 16, line 61) to be unpredictable (unlikely to be predicted accurately, col. 16, lines 61-62).

31. Sharangpani et al. do not teach that the branch instruction is determined by the compiler to be unpredictable but uses hardware means for the same (col. 7, lines 60-67;col. 8, lines 1-58).

32. Maki teaches that the compiler can determine the branch prediction information (col. 9, lines 25-28).

33. It would have been obvious to one of ordinary skill in the art at the time of the applicant's invention to have used Maki's teachings and modified the method of Sharangpani et al. by using the compiler to determine if a the branch instruction is unpredictable and hence reduce hardware.

34. In regard to Claim 5:

35. Sharangpani et al. teach to fetch instructions from the branch-taken (target) and branch-not-taken (sequential or second target) paths (col. 4, lines 13-15) but do not specifically mention to fetch a predetermined number of instructions. However, it is deemed to be inherent while fetching instructions to fetch a predetermined number of instructions.

36. In regard to Claim 6:

37. Sharangpani et al. teach to fetch instructions from the branch-not-taken (sequential or second target) path (col. 4, lines 13-15). However they do not specifically mention to fetch down the branch-not-taken path until a subsequent branch instruction is encountered. It would have been obvious to one of ordinary skill in the art at the time of the invention to fetch down the branch-not-taken path until a subsequent branch instruction is encountered because this follows normal instruction fetch semantics.

38. In regard to Claim 7:

39. Sharangpani et al. teach a microprocessor (Fig. 3, element 101) comprising:

1) an instruction cache (Fig. 3, element 301); and

2) a fetch unit (Fig. 3, element 304) connected to the instruction cache;

wherein the fetch unit is configured to speculatively fetch instructions (col. 5, lines 35-37) from both a branch-taken path and branch-not-taken path of the branch instruction (col. 16, lines 59-60) depending on the state of the branch prediction information (col. 16, lines 60-62) from a branch prediction unit (Fig. 3, element 336) for a branch instruction retrieved from the instruction cache.

40. Sharangpani et al. do not teach of a fetch unit, which is configured to detect branch instruction information in a branch instruction.

41. Maki teaches a prediction means (col. 12, line 65) within a fetch unit (prefetch circuit, col. 12, line 64), which detects (extracts, col. 12, line 65) branch instruction information (branch prediction bit, col. 12, line 65-67, branch destination address, col. 12, lines 51-52) in the branch instruction.

42. One of ordinary skill in the art at the time of the invention would have motivated to use the prediction means taught by Maki by detecting branch instruction information in the branch instruction instead of having branch prediction logic (Fig. 3, element 316) to generate branch instruction information as in Sharangpani et al. to reduce hardware costs.

43. It would have been obvious to one of ordinary skill in the art at the time of the applicant's invention to configure the fetch unit taught by Sharangpani et al. to detect branch instruction information in a branch instruction retrieved from the instruction cache to reduce hardware costs.

44. In regard to Claim 8:

45. Sharangpani et al. teach a branch prediction unit (Fig. 3, element 316) enabled by the branch instruction information (col. 13, lines 40-42, 46-48; Fig 6A, steps 610, 612) configured to predict the result of the branch instruction.

46. In regard to Claim 9:

47. Sharangpani et al. disclose a prediction bypass unit (fetch unit, Fig. 3, element 304), which is enabled by the branch prediction information (col. 9, lines 6-10) and configured to issue instruction addresses (fetches, col. 9, line 12) from a branch-taken path and a branch-not-taken path of the branch instruction (col. 9, lines 10-13).

48. In regard to Claim 10:

49. Sharangpani et al. further teach:

1) speculatively executing the instructions from the branch-taken path and the branch-not-taken path of the branch instruction (Fig. 6B, element 627, col. 14, lines 39-43);

2) execute (resolving, col. 17, line 37) the conditional branch instruction; and

3) based upon the outcome of the conditional branch instruction, discard the results from the speculatively executed instructions from the branch-taken path if the branch is not taken and discarding results from the branch-not-taken path if the branch is taken (col. 17, lines 38-41, 46-47).

50. In regard Claim 11:

51. Sharangpani et al. do not teach receiving branch prediction information comprising of compiler generated information indicative of the context in which the conditional branch instruction is used.

52. However Maki teaches that the branch prediction information (branch prediction bit, col. 9, line 24) is prepared according to the known prediction algorithm when a compiler makes a program (col. 9, lines 25-28) and is received from the branch instruction fetched (col. 12, lines 65-67).

53. On of ordinary skill in the art at the time of the invention would have recognized that by receiving the branch prediction information comprising of compiler generated information indicative of the context in which the conditional branch instruction as taught by Maki in place of the dynamic branch prediction (Fig. 3, element 316) method taught by Sharangpani et al. would have reduced costs by reducing prediction logic in hardware.

54. It would have been obvious to one of ordinary skill in the art at the time of the applicant's invention to have received branch prediction information comprising compiler generated information as taught by Maki for the branch prediction disclosed in Sharangpani et al to reduce hardware costs.

55. In regard to Claim 12:

56. Sharangpani et al. teach to fetch instructions from the branch-taken (target) and branch-not-taken (sequential or second target) paths (col. 4, lines 13-15) depending upon the state of the branch instruction information (branch prediction information) but do not specifically mention to fetch a predetermined number of instructions. However, it would have been obvious to one of ordinary skill in the art at the time of the applicant's invention to fetch a predetermined number of instructions while fetching instructions because this follows normal instruction fetch semantics.

57. In regard to Claim 13:

58. Sharangpani et al. teach to fetch instructions from the branch-not-taken (sequential or second target) path (col. 4, lines 13-15) depending upon the state of the branch instruction information (branch prediction information). However they do not specifically mention to fetch down the branch-not-taken path until a subsequent branch instruction is encountered. It would have been obvious to one of ordinary skill in the art at the time of the invention to fetch down the branch-not-taken path until a subsequent branch instruction is encountered because this follows normal instruction fetch semantics.

59. In regard to Claim 14:

60. Sharangpani et al. teach a data processing system (Fig. 1) including processor (element 101), memory (element 102), input means (element 112), and display (element 110), the processor including:

1) an instruction cache (Fig. 3, element 301); and

2) a fetch unit (Fig. 3, element 304) connected to the instruction cache;

wherein the fetch unit is configured to speculatively fetch instructions (col. 5, lines 35-37) from both a branch-taken path and branch-not-taken path of the branch instruction (col. 16, lines 59-60) depending on the state of the branch prediction information (col. 16, lines 60-62) from a branch prediction unit (Fig. 3, element 336) for a branch instruction retrieved from the instruction cache.

61. Sharangpani et al. do not teach of a fetch unit, which is configured to detect branch instruction information in a branch instruction.

62. Maki teaches a prediction means (col. 12, line 65) within a fetch unit (prefetch circuit, col. 12, line 64), which detects (extracts, col. 12, line 65) branch instruction information (branch prediction bit, col. 12, line 65-67, branch destination address, col. 12, lines 51-52) in the branch instruction.

63. One of ordinary skill in the art at the time of the invention would have motivated to use the prediction means taught by Maki by detecting branch instruction information in the branch instruction instead of having branch prediction logic (Fig. 3, element 316) to generate branch instruction information as in Sharangpani et al. to reduce hardware costs.

64. It would have been obvious to one of ordinary skill in the art at the time of the applicant's invention to configure the fetch unit taught by Sharangpani et al. to detect branch instruction information in a branch instruction retrieved from the instruction cache to reduce hardware costs.

65. In regard to Claim 15:

66. Sharangpani et al. teach a branch prediction unit (Fig. 3, element 316) enabled by the branch instruction information (col. 13, lines 40-42, 46-48; Fig 6A, steps 610, 612) configured to predict the result of the branch instruction.

67. In regard to Claim 16:

68. Sharangpani et al. disclose a prediction bypass unit (fetch unit, Fig. 3, element 304), which is enabled by the branch prediction information (col. 9, lines 6-10) and configured to issue instruction addresses (fetches, col. 9, line 12) from a branch-taken path and a branch-not-taken path of the branch instruction (col. 9, lines 10-13).

69. In regard to Claim 17:

70. Sharangpani et al. further teach:

1) speculatively executing the instructions from the branch-taken path and the branch-not-taken path of the branch instruction (Fig. 6B, element 627, col. 14, lines 39-43);

2) execute (resolving, col. 17, line 37) the conditional branch instruction; and

3) based upon the outcome of the conditional branch instruction, discard the results from the speculatively executed instructions from the branch-taken path if the branch is not taken and discarding results from the branch-not-taken path if the branch is taken (col. 17, lines 38-41, 46-47).

71. In regard Claim 18:

72. Sharangpani et al. do not teach receiving branch prediction information comprising of compiler generated information indicative of the context in which the conditional branch instruction is used.

73. However Maki teaches that the branch prediction information (branch prediction bit, col. 9, line 24) is prepared according to the known prediction algorithm when a compiler makes a program (col. 9, lines 25-28) and is received from the branch instruction fetched (col. 12, lines 65-67).

74. On of ordinary skill in the art at the time of the invention would have recognized that by receiving the branch prediction information comprising of compiler generated information indicative of the context in which the conditional branch instruction as taught by Maki in place of the dynamic branch prediction (Fig. 3, element 316) method taught by Sharangpani et al. would have reduced costs by reducing prediction logic in hardware.

75. It would have been obvious to one of ordinary skill in the art at the time of the applicant's invention to have received branch prediction information comprising compiler generated information as taught by Maki for the branch prediction disclosed in Sharangpani et al to reduce hardware costs.

76. In regard to Claim 19:

77. Sharangpani et al. teach to fetch instructions from the branch-taken (target) and branch-not-taken (sequential or second target) paths (col. 4, lines 13-15) depending upon the state of the branch instruction information (branch prediction information) but do not specifically mention to fetch a predetermined number of instructions. However, it would have been obvious to one of ordinary skill in the art at the time of the applicant's invention to fetch a predetermined number of instructions while fetching instructions because this follows normal instruction fetch semantics.

Conclusion

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. Applicant is reminded that in amending in reply to a rejection of claims in an application or patent under reexamination, the applicant or patent owner must clearly point out the patentable novelty, which he or she thinks the claims present in view of the state of the art disclosed by the references cited or the objections made. The applicant or patent owner must also show how the amendments avoid such references or objections. See 37 CFR § 1.111.

a. An article by L. Gwennap titled “DanSoft develops VLIW design” in the February 1997 issue of Microprocessor Report, teaches of a processor which implements a multiple path execution model using prediction information from a static branch prediction method. The hardware dynamically decides whether dual path execution is applied using the prediction information in the opcode bits of the branch instruction.

b. Kimura et al. (US005511172A) teach a speculative execution processor which fetches dual paths on detecting an unpredictable branch instruction.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Amol V. Gole whose telephone number is 703-305-8888. The examiner can normally be reached on 8:30-5:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on 703-305-9712. The fax phone number for the organization where this application or proceeding is assigned is (703) 872-9306.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is 703-305-3900.

AVG

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100